RSAC 2025 Conference

Many Voices.
One Community.

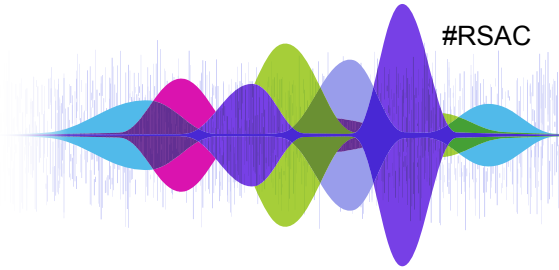
SESSION ID: APP-W02

525,600 Assessments Later Top Mobile App Risks Since 2022

Andrew Hoog

Co-Founder / Board Member
NowSecure, Inc.

andrewhoog



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

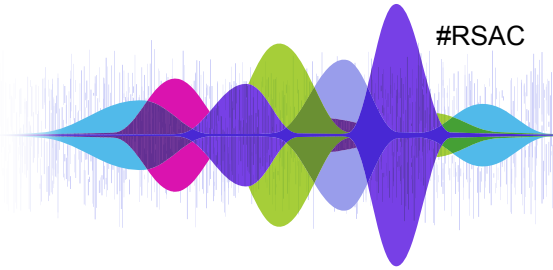
Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

RSAC™ | 2025 Conference

Introduction

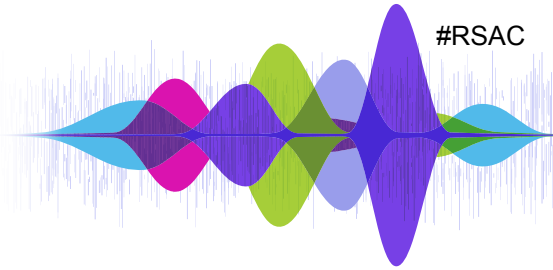
Mobile apps have infiltrated every facet of our lives



- ~7.21 billion smartphones in use globally
- In 2023, approximately 255 billion mobile app downloads globally (+25b from 2022)
- Average smartphone user has between 60 and 90 apps installed on their device

Mobile apps power our personal, business and civic lives.

Why mobile app flaws are hard to detect



1. Platform Restrictions

- Inability to have root or administrator access to the device / operating system
- App sandboxing
- Limited platform security APIs and visibility

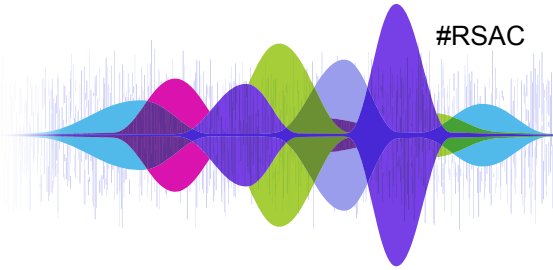
2. Attack Surface

- Entire web app attack surface as many apps instantiate in-app browser instances for auth or other use cases
- 18+ sensors like GPS, accelerometer, etc.
- Significant use of open and closed source 3rd party SDKs

3. Frameworks & Languages

- Yearly Android and iOS updates change how core OS frameworks operate
- Newer languages like Swift and Kotlin have less support from security tools

Why does this matter



- Attackers are collecting swaths of sensitive data from mobile apps with security and privacy issue
- Defenders lack of visibility into device security telemetry and limited access for researchers creates an asymmetric security environment where attackers can exploit vulnerabilities at scale with little chance of detection.
- These flaws undermine our personal privacy and our national and economic security.

If you leave with 1 question …

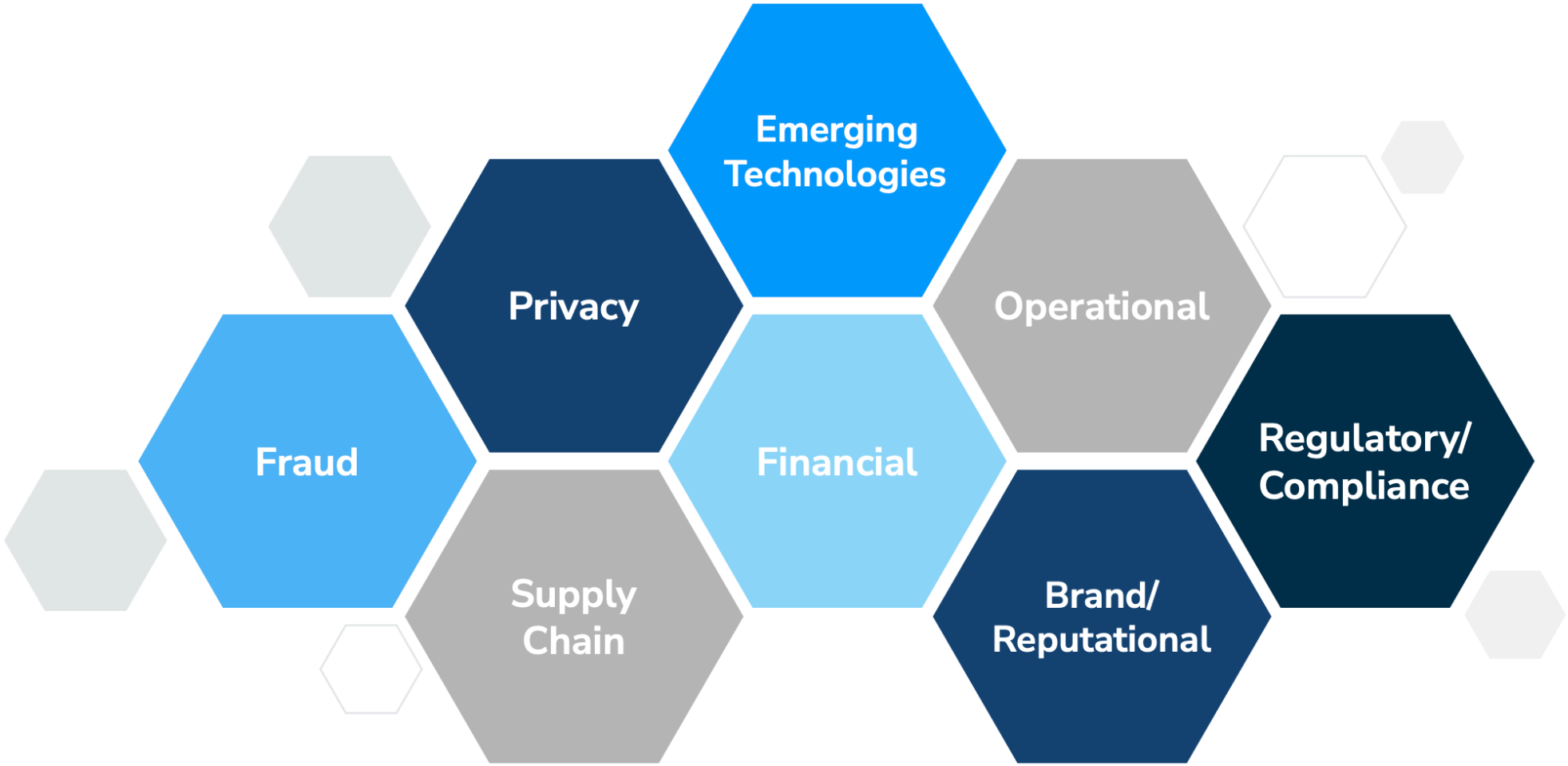
RSAC | 2025 Conference

Top Mobile App Risks

From 2022-01 to 2025-02

Many Voices.
One Community.

Types of Mobile Apps Risks



5. Reconnaissance - MASVS-RESILIENCE-3

Example: Debug Symbols Not Stripped

- List of symbols in your app
  - Class names
  - Global variables
  - Method and function names
- Allow reverse engineering of your app more quickly

Sample of Responsive Findings

Finding	Found
Debug Symbols Not Stripped	72.43%
Debug Symbols (DWARF) Not Stripped	70.57%
API Discovery	87.28%
SDKs and Frameworks	83.78%
Hardcoded URLs	67.72%

Real World Example - My Security Account App

1. EXECUTIVE SUMMARY

- **CVSS v4 8.7**
- **ATTENTION**: Exploitable remotely/low attack complexity
- **Vendor**: Rapid Response Monitoring
- **Equipment**: My Security Account App
- **Vulnerability**: Authorization Bypass Through User-Controlled Key

2. RISK EVALUATION

Successful exploitation of this vulnerability could allow attacker to access sensitive information of other users.

3. TECHNICAL DETAILS

3.1 AFFECTED PRODUCTS

The following Rapid Response Monitoring products are affected:

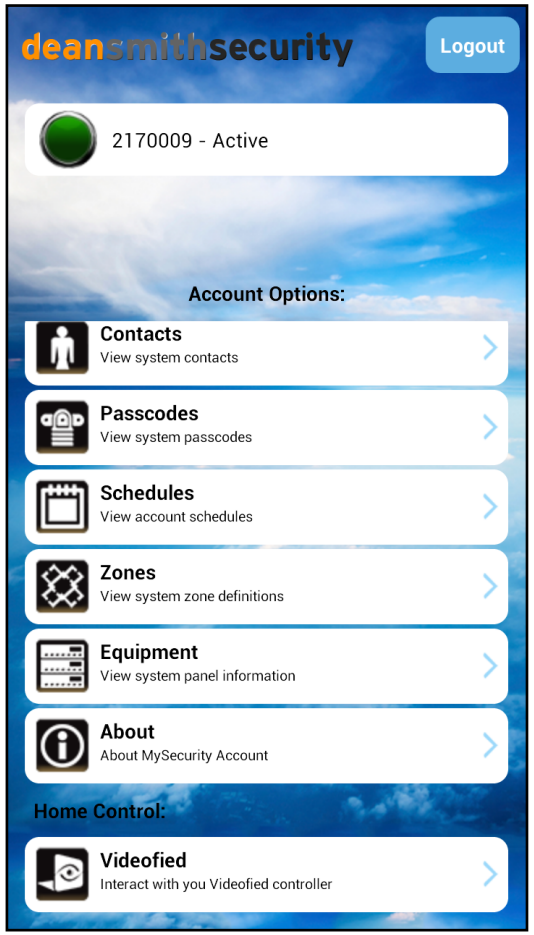
- My Security Account App API: Versions prior to 7/29/24

3.2 VULNERABILITY OVERVIEW

3.2.1 Authorization Bypass Through User-Controlled Key CWE-639

Rapid Response Monitoring My Security Account App utilizes an API that could be exploited by an attacker to modify request data, potentially causing the API to return information about other users.

CVE-2025-0352 has been assigned to this vulnerability. A CVSS v3.1 base score of 7.5 has been calculated; the CVSS vector string is (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N).



https://www.cisa.gov/news-events/ics-advisories/icsa-25-051-05

Real World Example - SparkCat

malicious executable. Unlike the Android version, the iOS framework retained debugging symbols, which allowed us to identify several unique details:

- The lines reveal the paths on the framework creators' device where the project was stored, including the user names:
  - **/Users/qiongwu/**: the project author's home directory
  - **/Users/quiwengjing/**: the Rust library creator's home directory
- The C2-rust communication module was named **im_net_sys**. Besides the client, it contains code that the attackers' server presumably uses to communicate with victims.
- The project's original name is **GZIP**.

https://securelist.com/sparkcat-stealer-in-app-store-and-google-play/115385/

4. Encryption flaws - MASVS-CRYPTO-1

Example: App is Encoding Sensitive Information Using Outdated or Insecure Cryptography

- Findings specifically focused on sensitive data
- Outdated or insecure crypto puts sensitive data at risk
- Looking ahead: PCQ

Sample of Responsive Findings

Finding	Found
App is Encoding Sensitive Information Using Outdated or Insecure Cryptography	61.77%
SSL Configuration Allows Insecure Connections	30.06%
Initialization Vector Reused for Encryption	31.56%
App Uses Static Values for Cryptography Which Exposes Data to Easy Decryption	20.47%

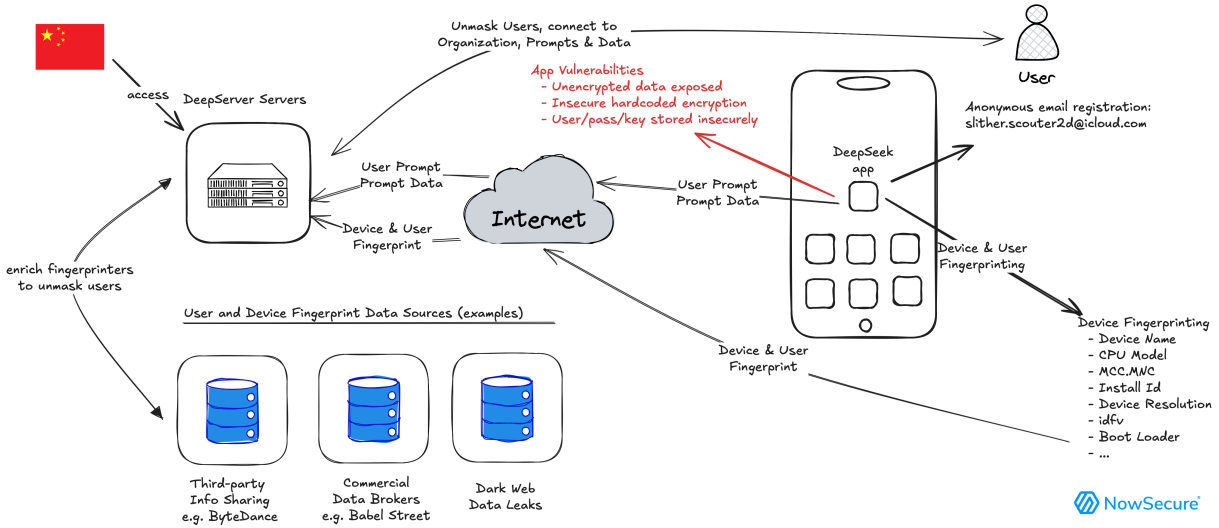
Real World Example - DeepSeek iOS

Executive Summary

NowSecure has conducted a comprehensive security and privacy assessment of the DeepSeek iOS mobile app, uncovering multiple critical vulnerabilities that put individuals, enterprises, and government agencies at risk. These findings highlight the immediate need for organizations to prohibit the app's use to safeguard sensitive data and mitigate potential cyber risks.

Key Risks Identified:

1. **Unencrypted Data Transmission:** The app transmits sensitive data over the internet without encryption, making it vulnerable to interception and manipulation.
2. **Weak & Hardcoded Encryption Keys:** Uses outdated Triple DES encryption, reuses initialization vectors, and hardcodes encryption keys, violating best security practices.
3. **Insecure Data Storage:** Username, password, and encryption keys are stored insecurely, increasing the risk of credential theft.
4. **Extensive Data Collection & Fingerprinting:** The app collects user and device data, which can be used for tracking and de-anonymization.
5. **Data Sent to China & Governed by PRC Laws:** User data is transmitted to servers controlled by ByteDance, raising concerns over government access and compliance risks.



https://www.nowsecure.com/blog/2025/02/06/nowsecure-uncovers-multiple-security-and-privacy-flaws-in-deepseek-ios-mobile-app/

3. Untested 3rd Party SDKs - MASVS-CODE-3

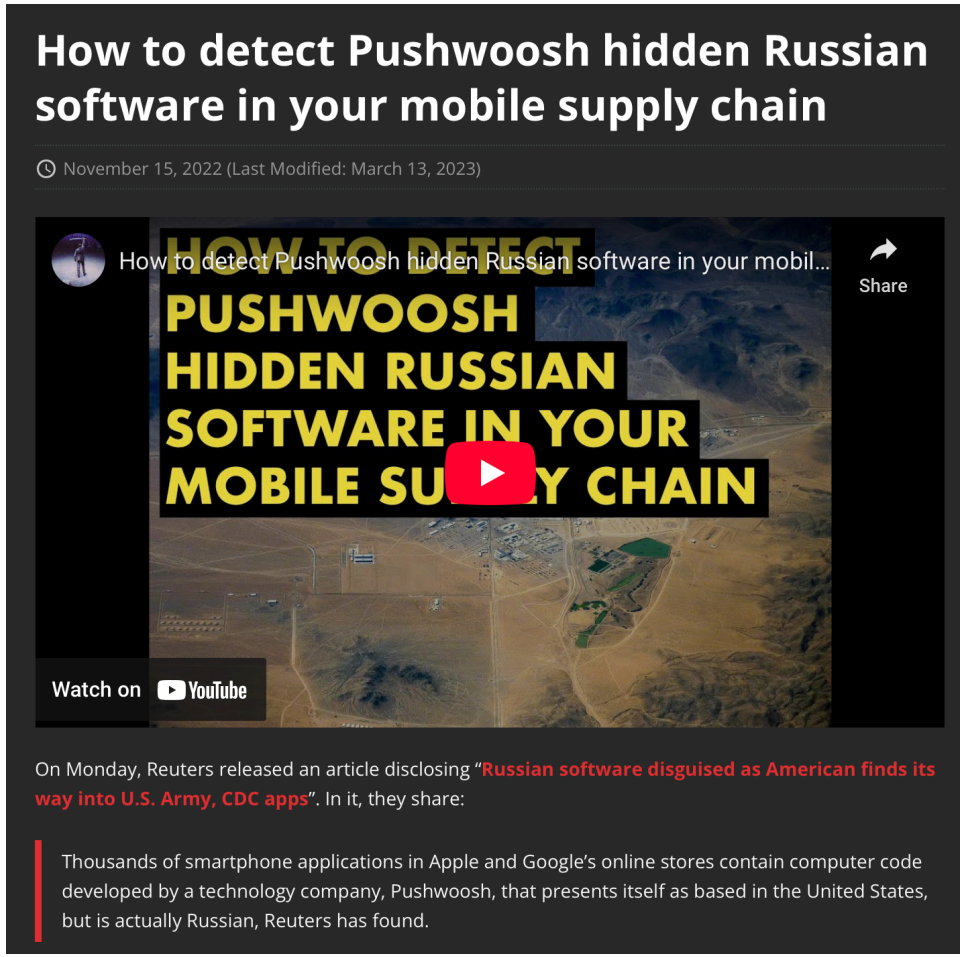
Example: Components with Known Vulnerabilities

- 60%+ of mobile apps are 3rd Party SDK
- The source code of many SDKs is not available for static source scans
- Very few SDK vulnerabilities are known and have CVEs assigned
- But we still find in >15% of apps

Sample of Responsive Findings

Finding	Found
Components with Known Vulnerabilities	15.77%
Outdated libpng Library Contains Known Security Flaw	13.11%
Misconfigured AFNetworking Library Potentially Allows Network Traffic Interception and Modification	5.20%
SDKs and Frameworks	83.78%
Hardcoded URLs	67.72%

Real World Example - Pushwoosh SDK


How to detect Pushwoosh hidden Russian software in your mobile supply chain
November 15, 2022 (Last Modified: March 13, 2023)
How to detect Pushwoosh hidden Russian software in your mobil...
Share
HOW TO DETECT
PUSHWOOSH
HIDDEN RUSSIAN
SOFTWARE IN YOUR
MOBILE SU Y CHAIN
Watch on YouTube
On Monday, Reuters released an article disclosing "Russian software disguised as American finds its way into U.S. Army, CDC apps". In it, they share:
Thousands of smartphone applications in Apple and Google's online stores contain computer code developed by a technology company, Pushwoosh, that presents itself as based in the United States, but is actually Russian, Reuters has found.

Key findings

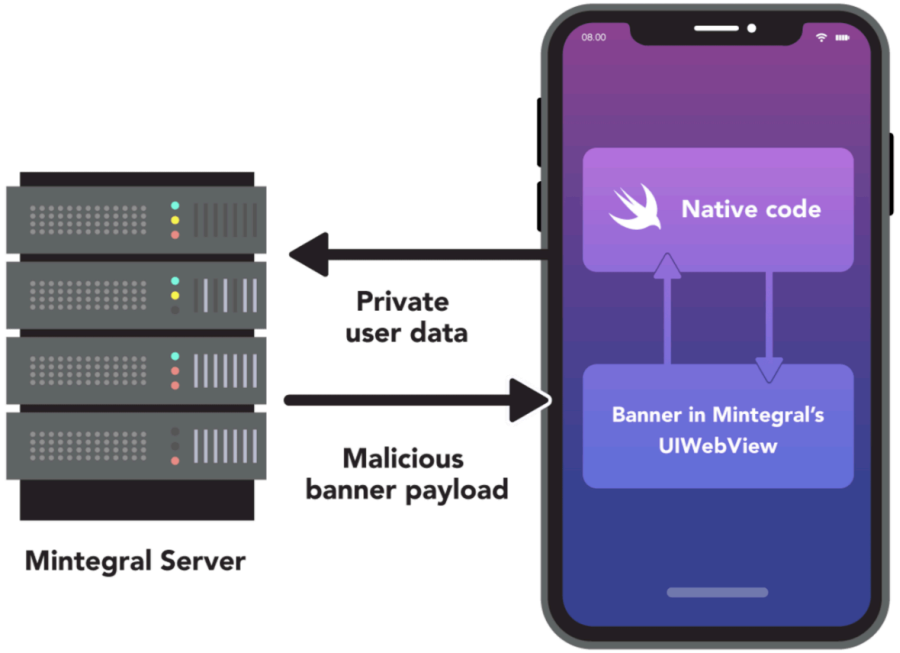
- More than 2.3 billion devices listed in their tracking database.
- Presents itself as based in the United States, but is actually Russian
- Pushwoosh used LinkedIn accounts purportedly belonging to two Washington, D.C.-based executives named Mary Brown and Noah O'Shea to solicit sales. But neither Brown nor O'Shea are real people, Reuters found.

https://www.andrewhoog.com/post/how-to-detect-pushwoosh-hidden-russian-software-in-your-mobile-supply-chain/

Real World Example - SourMint malicious SDK

This writeup is divided into three sections:

- Excessive Data Collection, including all HTTP request interception and logging in the iOS SDK - originally published on 24th of August 2020 - describes the URL and request tracking capabilities in the iOS distribution of the Mintegral SDK.
- A backdoor in the iOS distribution of the SDK allows Remote Code Execution - describes the remote code execution capabilities in MintegralAdSDK published on 15th of October 2020.
- Downloads URL tracking in Android - describes various findings in the Android distribution of the Mintegral SDK.



https://snyk.io/blog/sour-mint-malicious-sdk/

2. Security Misconfigurations - MASVS-PLATFORM-1

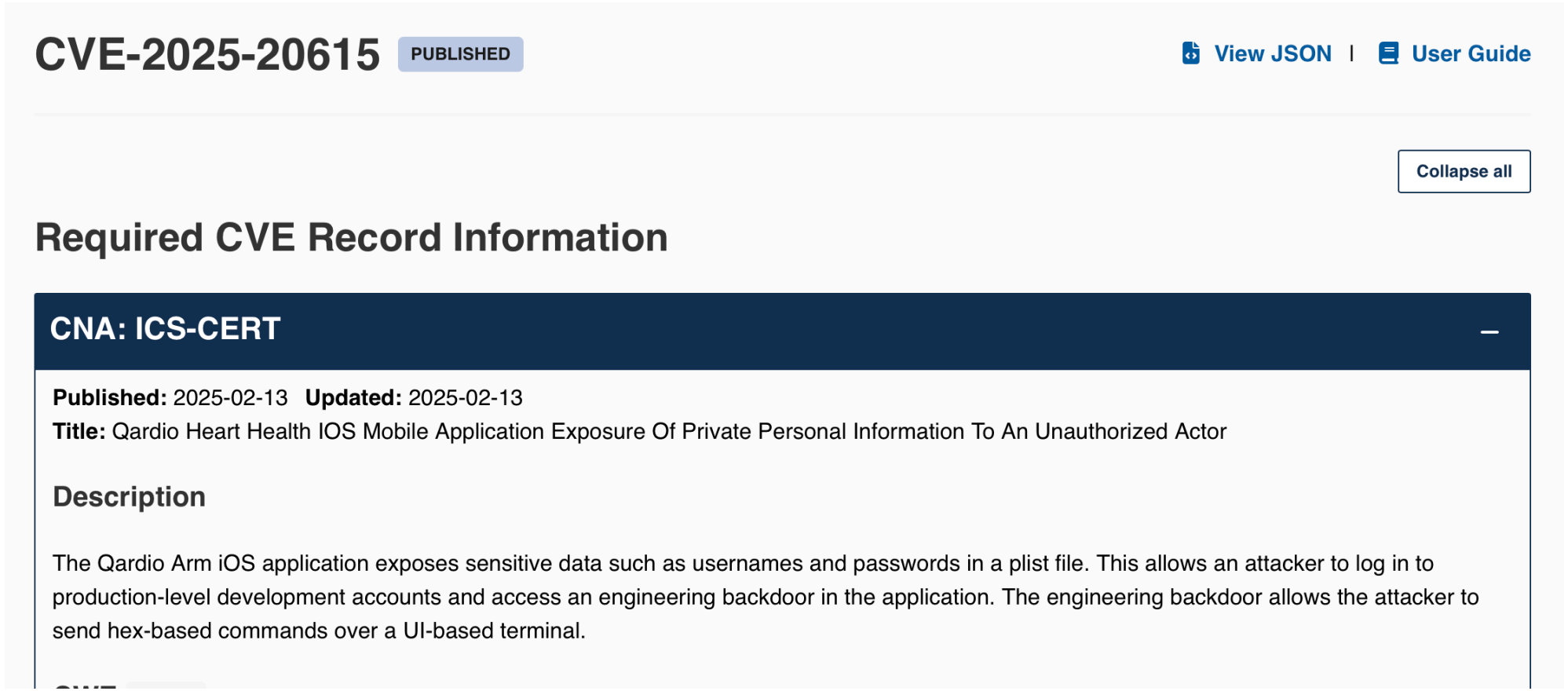
Example: Hardcoded Cryptographic Keys Found in Use

- Hardcoded keys are trivial to extract
- Sensitive data is at risk when hard coded keys are used
- If hardcoded keys are

Sample of Responsive Findings

Finding	Found
Context Registered Broadcast Receivers Not Protected with Permissions	96.38%
App Uses Custom URL Schemes Registered Which Are Susceptible to Hijacking	79.62%
Hardcoded Cryptographic Keys Found in Use	19.28%
Hardcoded AI API Keys	0.12%

Real World Example - Qardio Health iOS



CVE-2025-20615 PUBLISHED

View JSON | User Guide

Collapse all

Required CVE Record Information

CNA: ICS-CERT

Published: 2025-02-13 **Updated:** 2025-02-13

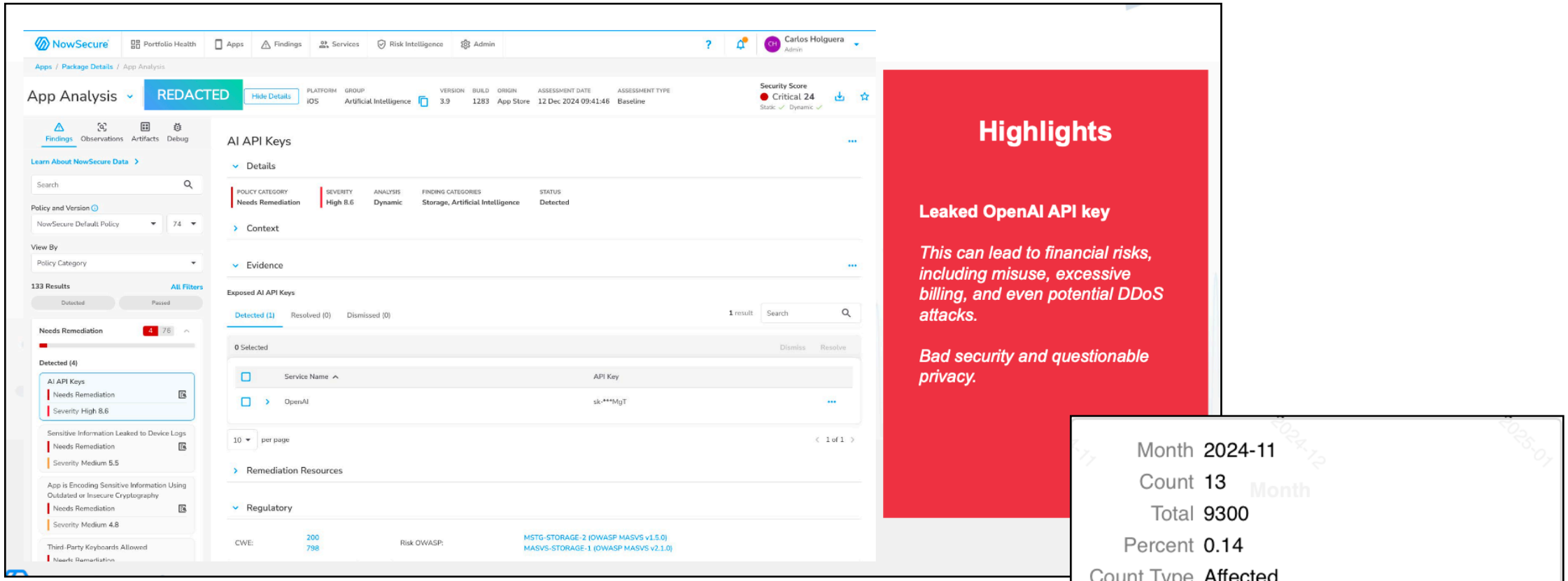
Title: Qardio Heart Health IOS Mobile Application Exposure Of Private Personal Information To An Unauthorized Actor

Description

The Qardio Arm iOS application exposes sensitive data such as usernames and passwords in a plist file. This allows an attacker to log in to production-level development accounts and access an engineering backdoor in the application. The engineering backdoor allows the attacker to send hex-based commands over a UI-based terminal.

https://www.cve.org/CVERecord?id=CVE-2025-20615

Real World Example - Leaked OpenAI Key



https://www.nowsecure.com/blog/2025/02/26/ai-risks-in-mobile-apps-how-to-protect-your-data-and-stay-compliant/

1. Privacy issues - MASVS-PRIVACY

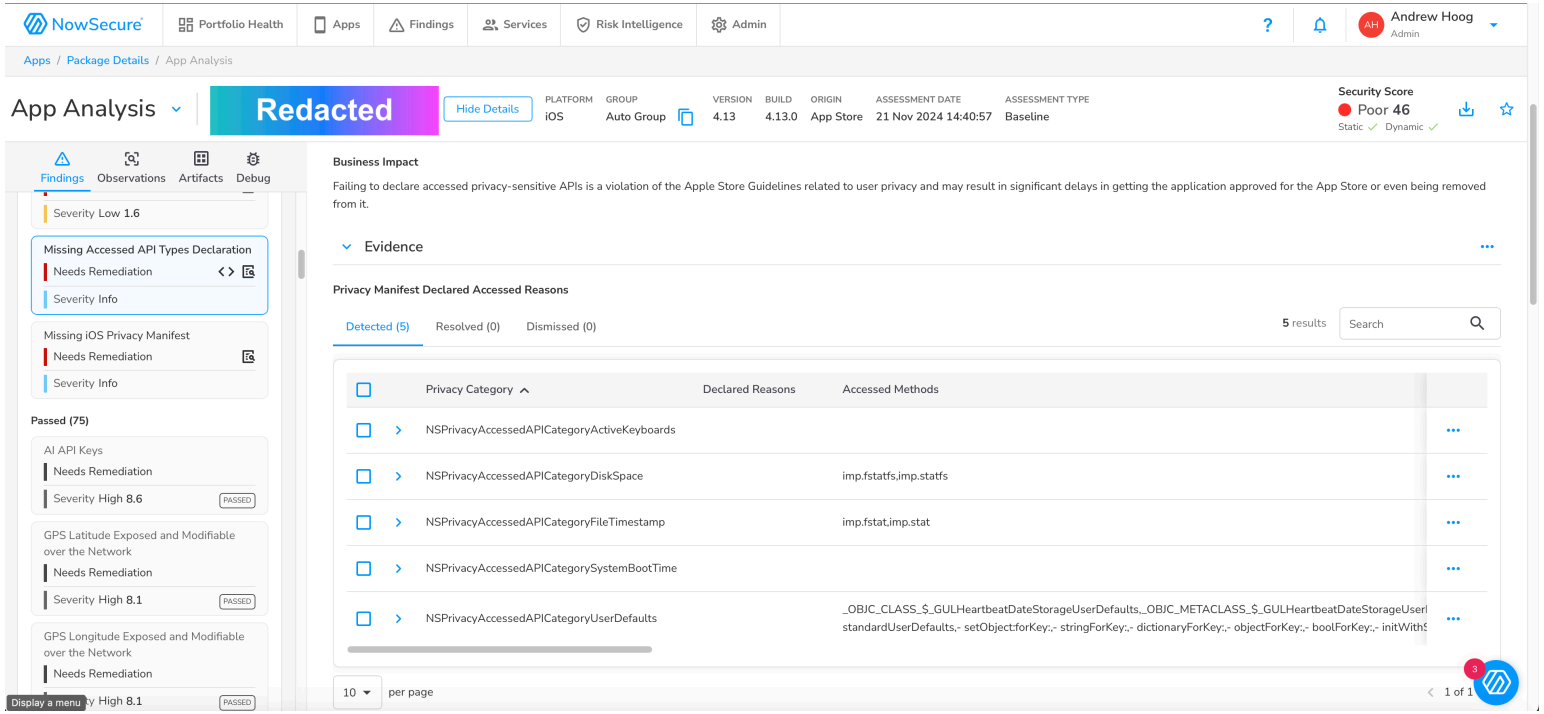
Example: Unique Device Identifier (Build Fingerprint) Stored Insecurely on Device

- Many 3rd party SDKs log this
- Often shrugged off as not privacy impacting
- However, identifier often contains owner's first and last name ("Andrew Hoog iPad")
- Not only GDPR/CCPA issues but if user is < 13, COPPA

Sample of Responsive Findings

Finding	Found
Missing Purpose Strings (ios)	75.75%
Potentially Sensitive Data Found in HTTP Cache Databases	40.49%
HTTP Requests in Network Traffic	9.55%
Failure to Validate Certificate Authority Allows Data to be Exposed and Modified Remotely	2.97%

Real World Example - Missing Privacy Manifests App Store Blocker

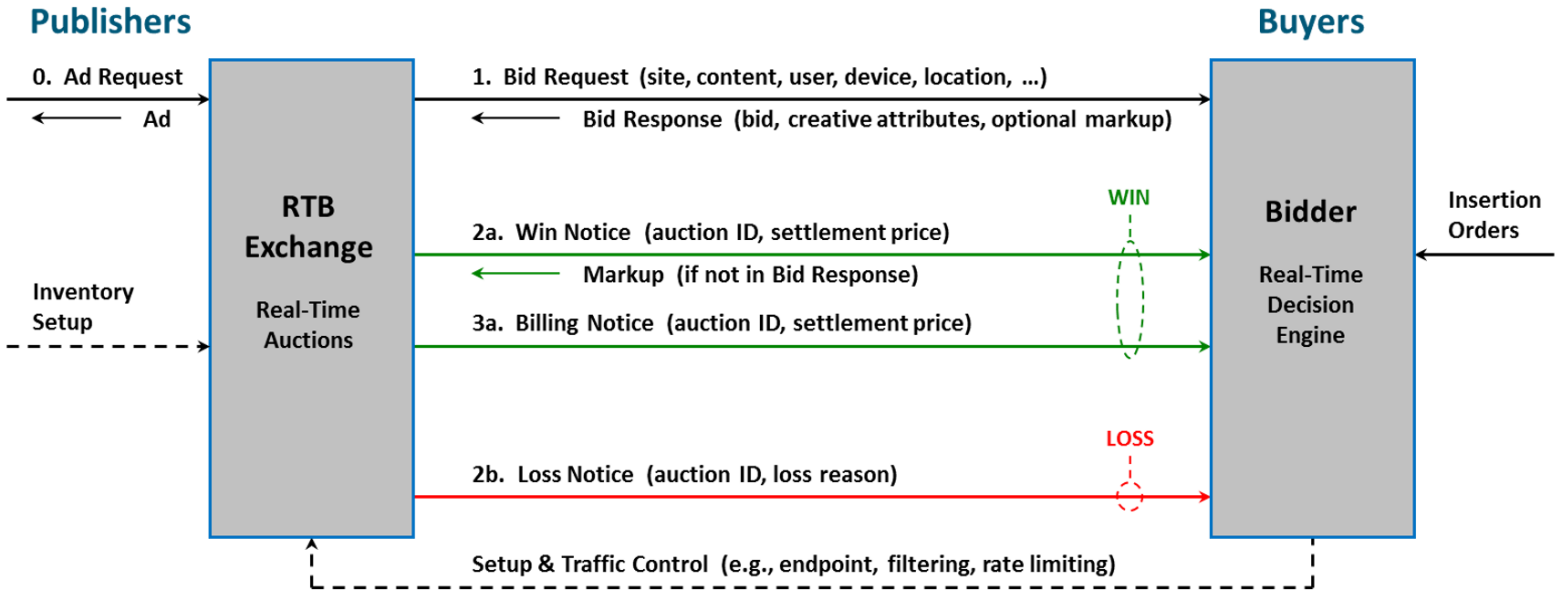


Privacy Attestations

Both Apple and Google are requiring and growing their requirements for Privacy Attestations.

Failure to comply can lead to delayed releases of app updates, impacting the business.

Real World Example - OpenRTB



2.2 - Security

HTTPS is not technically required for a functioning OpenRTB implementation. However, HTTPS is increasingly customary. It is strongly recommended that exchanges and bidders use only HTTPS.

https://github.com/InteractiveAdvertisingBureau/openrtb2.x/blob/main/2.6.md#22---security-

Impact of App Tracking Transparency

APPLES AND GOOGLES

Atlas found that for the right price (typically $10-50k a year), brokers can provide access to tens of billions of data points covering large swaths of the US population and the rest of the world.

Based on the data sets Atlas acquired — many of which included older MAID records — they estimate they could locate roughly 80 percent of Android-based devices and about 25 percent of Apple phones. Google refers to its MAID as the "**Android Advertising ID**," (AAID) while Apple calls it the "**Identifier for Advertisers**" (IDFA).

What accounts for the disparity between the number of Android and Apple devices that can be found in mobile advertising data? In April 2021, Apple shipped version 14.5 of its iOS operating system, which introduced a technology called **App Tracking Transparency** (ATT) that requires apps to get affirmative consent before they can track users by their IDFA or any other identifier.

Apple's introduction of ATT had a swift and profound impact on the advertising market: Less than a year later Facebook disclosed that the iPhone privacy feature would decrease the company's 2022 revenues by about $10 billion.

TECH

Facebook says Apple iOS privacy change will result in $10 billion revenue hit this year

PUBLISHED WED, FEB 2 2022·7:54 PM EST | UPDATED THU, FEB 3 2022·11:06 AM EST

Kif Leswing
@KIFLESWING

SHARE

Source: cnbc.com.

Key findings

- Investigation by Atlas Data Privacy Corp. on Babel Street's Locate X Platform
- Could locate roughly 80% of Android device but "only" 25% of iOS devices
- Apple's App Tracking Transparency has severely curbed tracking of iOS devices

https://krebsonsecurity.com/2024/10/the-global-surveillance-free-for-all-in-mobile-ad-data/

Gravy Analytic Breach

Breach Details

- 17TB of data breached from Gravy Analytics' AWS cloud storage.
- Exposed precise location data revealing individuals' movements, residences, and workplaces.
- Data sourced from thousands of apps, including Tinder, Grindr, Candy Crush, and MyFitnessPal.

JOSEPH COX SECURITY JAN 9, 2025 4:05 PM

Candy Crush, Tinder, MyFitnessPal: See the Thousands of Apps Hijacked to Spy on Your Location

A hack of location data company Gravy Analytics has revealed which apps are—knowingly or not—being used to collect your information behind the scenes.

If the product is free …

NowSecure®

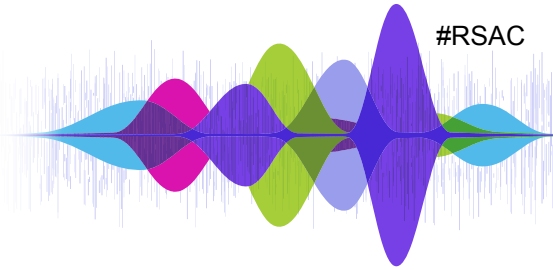
Why is this happening?

1. Incorrect assumption (even by security professionals) that believe the public mobile app stores provide sufficient app vetting for mobile apps. MYTH
2. Companies understandably worry about they apps they build but overlook the 100s/1000s of apps they use that have access to sensitive data
3. Many organizations test mobile apps with web testing techniques (static source code analysis) which misses the most of the mobile attack surface including 3rd party code and most runtime issues found with dynamic.

RSAC™ | 2025 Conference

Apply It

So what can you do?



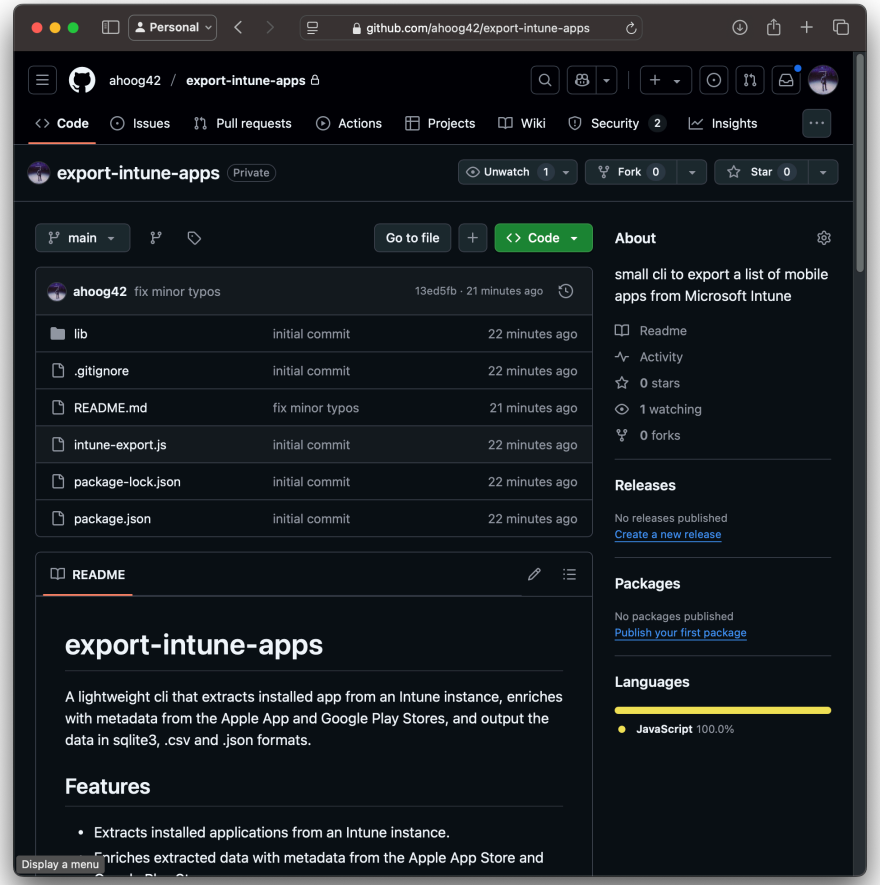
- Good news is you can quickly being to address this risk leveraging open source tools
- While the techniques will be bespoke for mobile, the approach follows the standard best practices for cybersecurity
- The following slides will walk you through the steps you can take to address mobile app risk

Next week you should

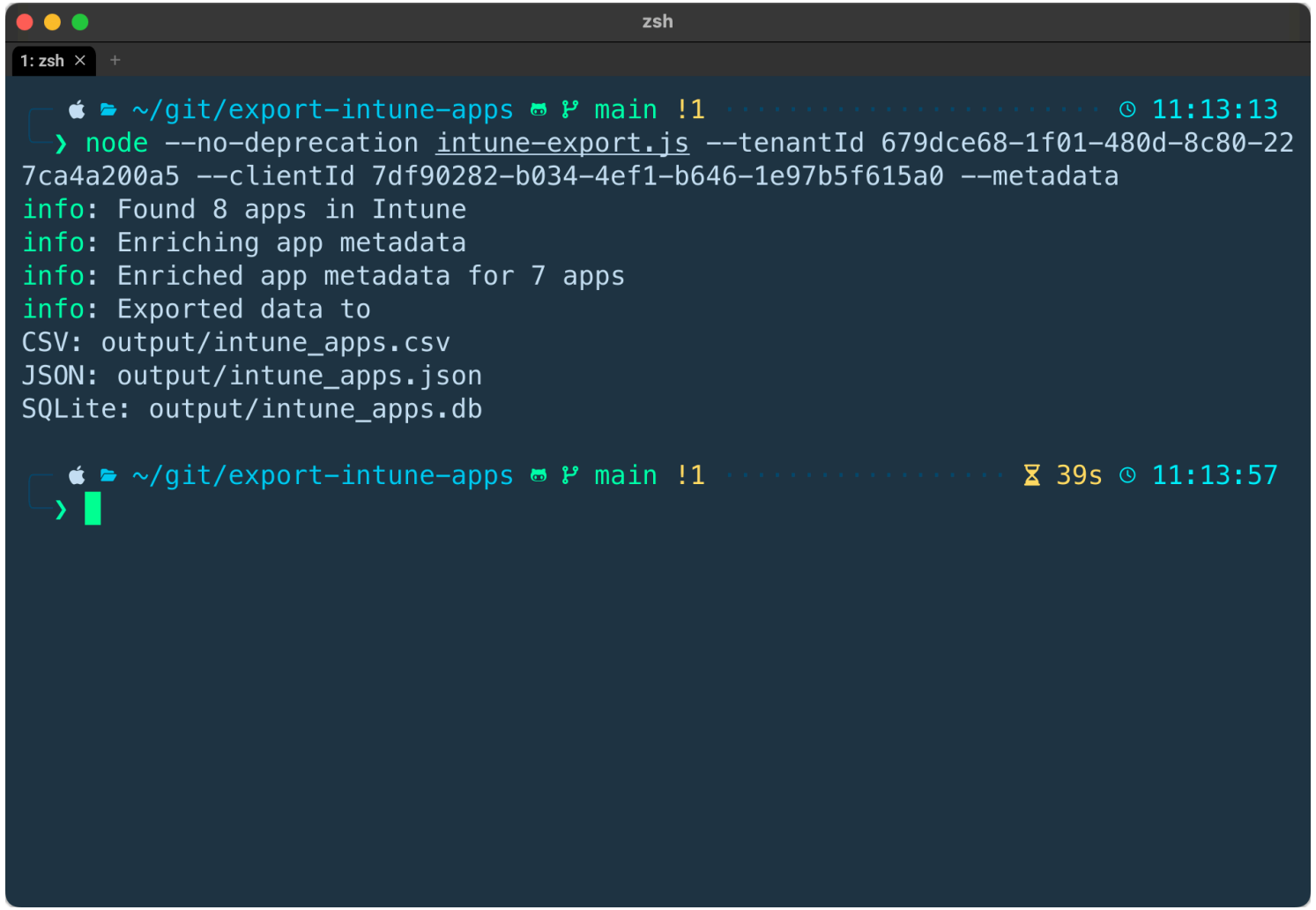
1. Inventory the apps your company uses or build
2. Place apps into Risk Profiles, e.g. “No sensitive data”, “PII”, “Flagship”

I’ve open sourced export-intune-apps, a simple command line tool that:

- exports an inventory of apps from Microsoft Intune
- enrich the app list with metadata from App/Play stores
- exports the data to csv, json and sqlite3



Demo: extract-intune-apps



In the first three months you should:

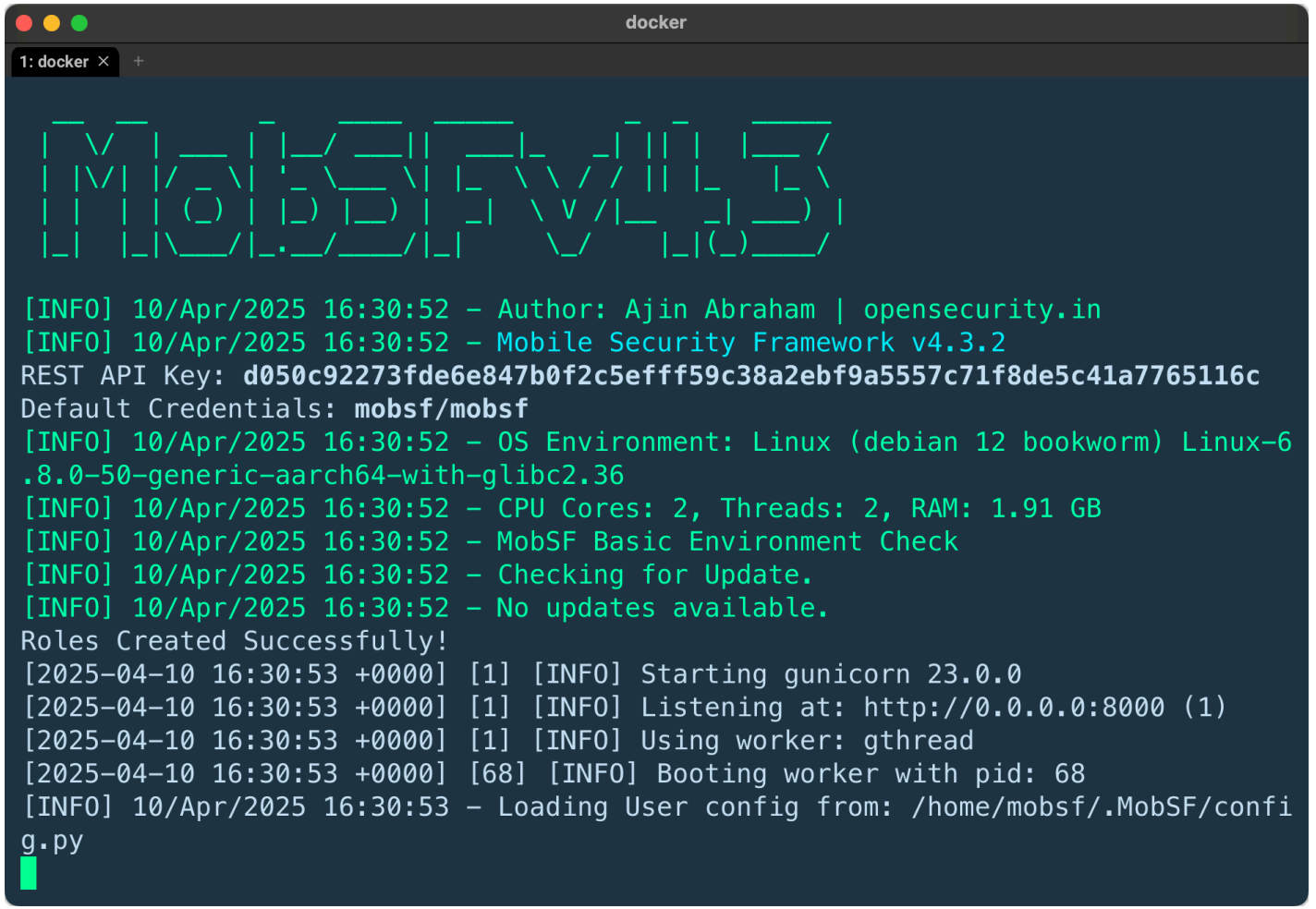
1. Review OWASP's MASVS and identify key mobile app tests needed to address your organization's risk profiles
2. Test mobile apps. I will demo the open sourced Mobile Security Framework (mobSF) from Ajin Abraham | opensecurity.in

mobSF how to

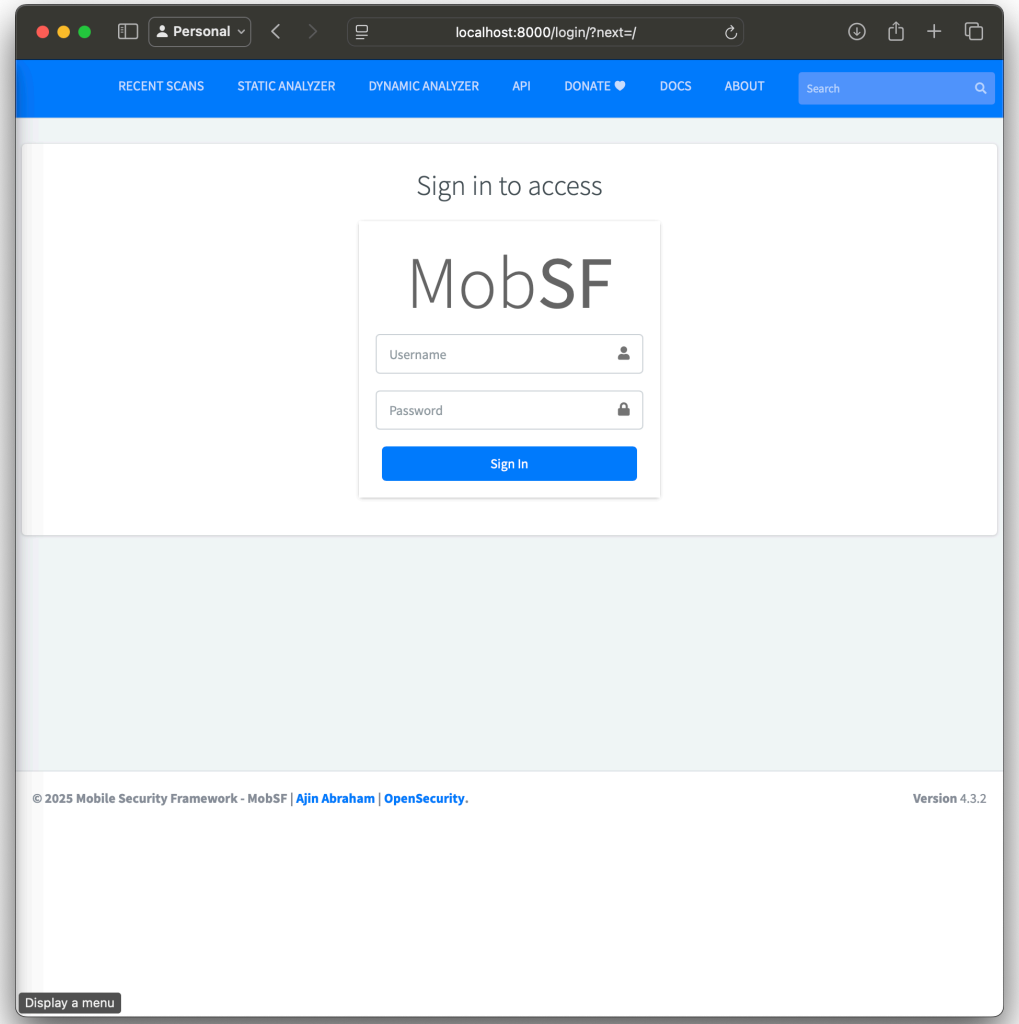
Let's use mobSF's docker image:

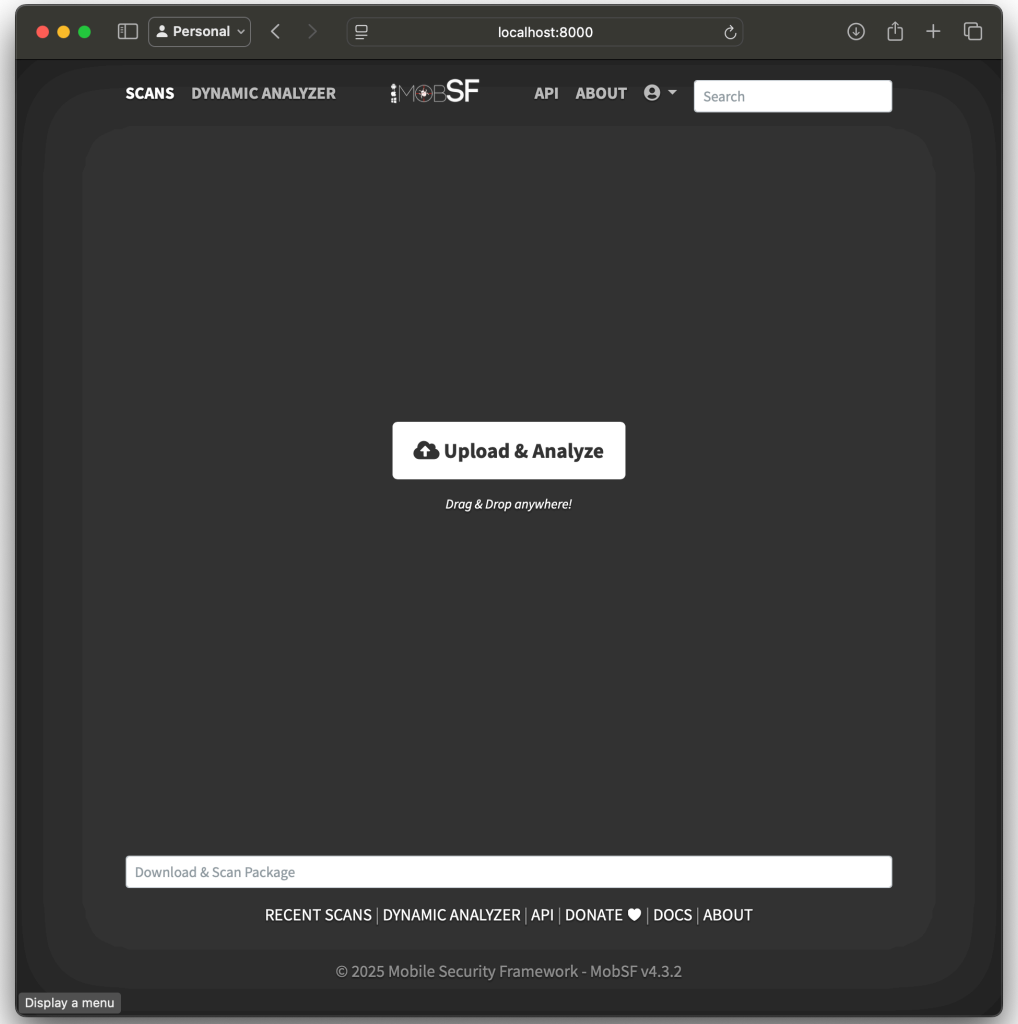
```
mkdir ~/mobsf

docker pull opensecurity/mobile-security-framework-mobsf:latest

docker run -it --rm --name mobsf -p 8000:8000 \
--mount type=bind,source=/Users/<username>/mobsf,target=/home/
mobsf/.MobSF opensecurity/mobile-security-framework-mobsf:latest
```

mobSF running in docker

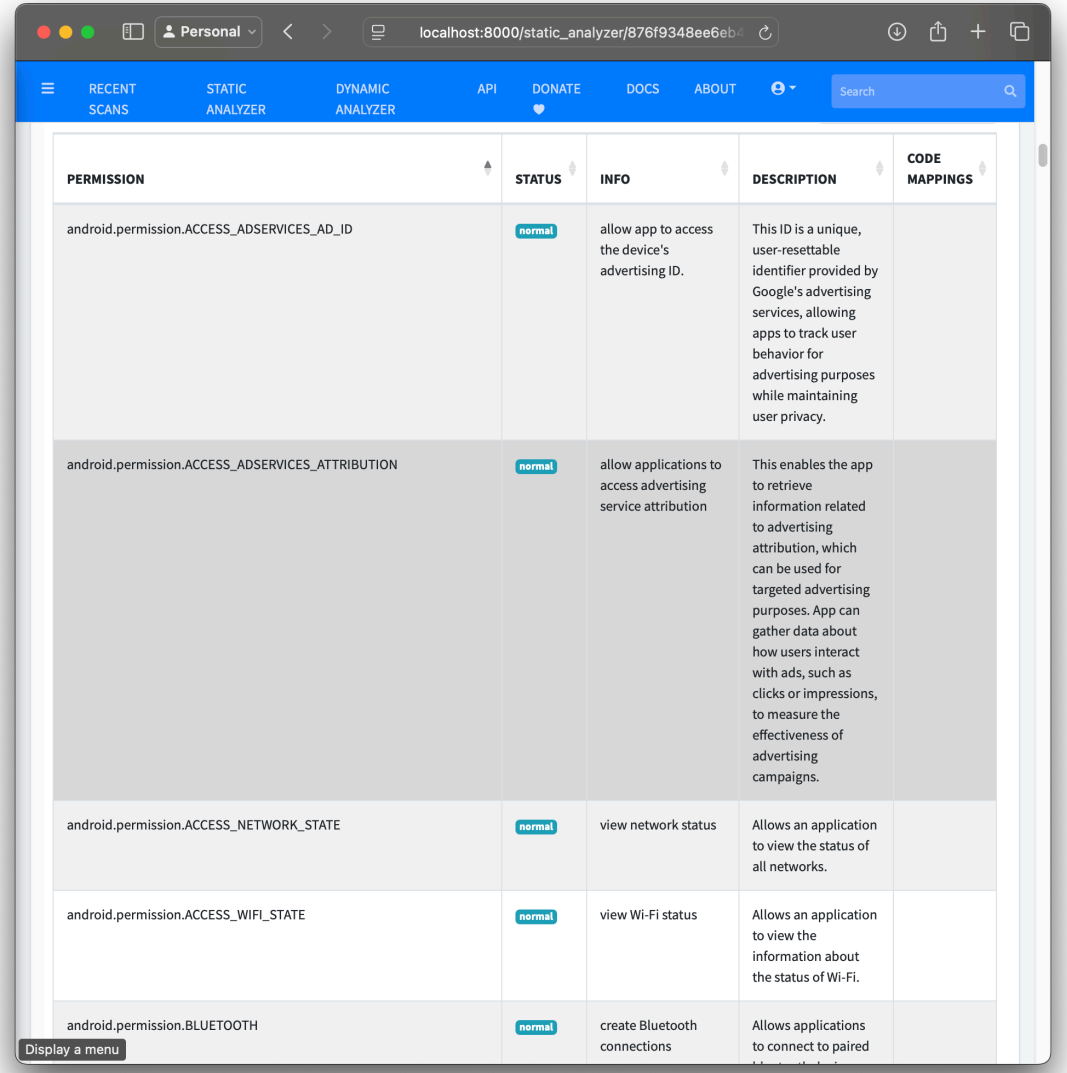

docker
1: docker
__ __ _ _____ _____ _ _ _____
| \/ | ___ | |__/ ___|| ___|_ _| || | |___ /
| |\/| |/ _ \| '_ ___ \| |_ \ \ / / || |_ |_ \
| | | | (_) | |_) |__) | _| \ V /|__ _| ___) |
|_| |_|___/|_.__/____/|_| _/ |_|(_)____/
[INFO] 10/Apr/2025 16:30:52 - Author: Ajin Abraham | opensecurity.in
[INFO] 10/Apr/2025 16:30:52 - Mobile Security Framework v4.3.2
REST API Key: d050c92273fde6e847b0f2c5efff59c38a2ebf9a5557c71f8de5c41a7765116c
Default Credentials: mobsf/mobsf
[INFO] 10/Apr/2025 16:30:52 - OS Environment: Linux (debian 12 bookworm) Linux-6.8.0-50-generic-aarch64-with-glibc2.36
[INFO] 10/Apr/2025 16:30:52 - CPU Cores: 2, Threads: 2, RAM: 1.91 GB
[INFO] 10/Apr/2025 16:30:52 - MobSF Basic Environment Check
[INFO] 10/Apr/2025 16:30:52 - Checking for Update.
[INFO] 10/Apr/2025 16:30:52 - No updates available.
Roles Created Successfully!
[2025-04-10 16:30:53 +0000] [1] [INFO] Starting gunicorn 23.0.0
[2025-04-10 16:30:53 +0000] [1] [INFO] Listening at: http://0.0.0.0:8000 (1)
[2025-04-10 16:30:53 +0000] [1] [INFO] Using worker: gthread
[2025-04-10 16:30:53 +0000] [68] [INFO] Booting worker with pid: 68
[INFO] 10/Apr/2025 16:30:53 - Loading User config from: /home/mobsf/.MobSF/config.py

Access web interface at localhost:8000

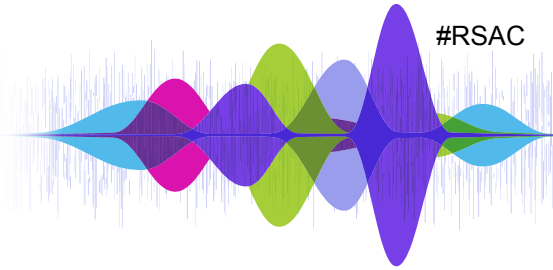




Results for Spotify Android



PERMISSION	STATUS	INFO	DESCRIPTION	CODE MAPPINGS
android.permission.ACCESS_ADSERVICES_AD_ID	normal	allow app to access the device's advertising ID.	This ID is a unique, user-resettable identifier provided by Google's advertising services, allowing apps to track user behavior for advertising purposes while maintaining user privacy.	
android.permission.ACCESS_ADSERVICES_ATTRIBUTION	normal	allow applications to access advertising service attribution	This enables the app to retrieve information related to advertising attribution, which can be used for targeted advertising purposes. App can gather data about how users interact with ads, such as clicks or impressions, to measure the effectiveness of advertising campaigns.	
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.	
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.	
android.permission.BLUETOOTH	normal	create Bluetooth connections	Allows applications to connect to paired	



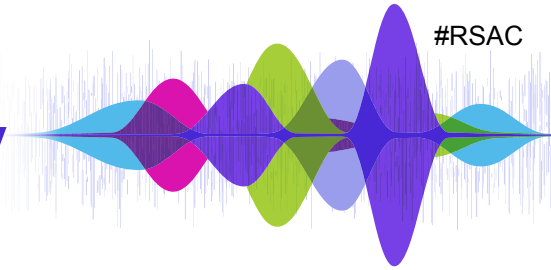
Within six months you should:

1. Review apps tested over past 3 months
2. Develop process to ensure scans are performed on each new app version (e.g. expand cli to submit apps to mobSF API and pull data from test identified in previous step)
3. Integrate results with mobile EDR/MTD/MDM and with enterprise-wide risk reporting

RSAC™ 2025 Conference

Summary

Mobile app risks are impacting you today



- Discovering mobile risks is difficult due to platform architecture
- This leads to few vulnerability researchers and tools to detect risks
- Yet mobile apps power our personal, business and civic lives
- Apple and Google do not rigorously testing mobile apps for security and privacy issues
- Enterprises continuously testing new mobile app releases for security and privacy issues will significantly reduce the impact to their organization

If you leave with 1 question …

RSAC™ 2025 Conference

Many Voices.
One Community.

Andrew Hoog

@ahoog42

andrewhoog